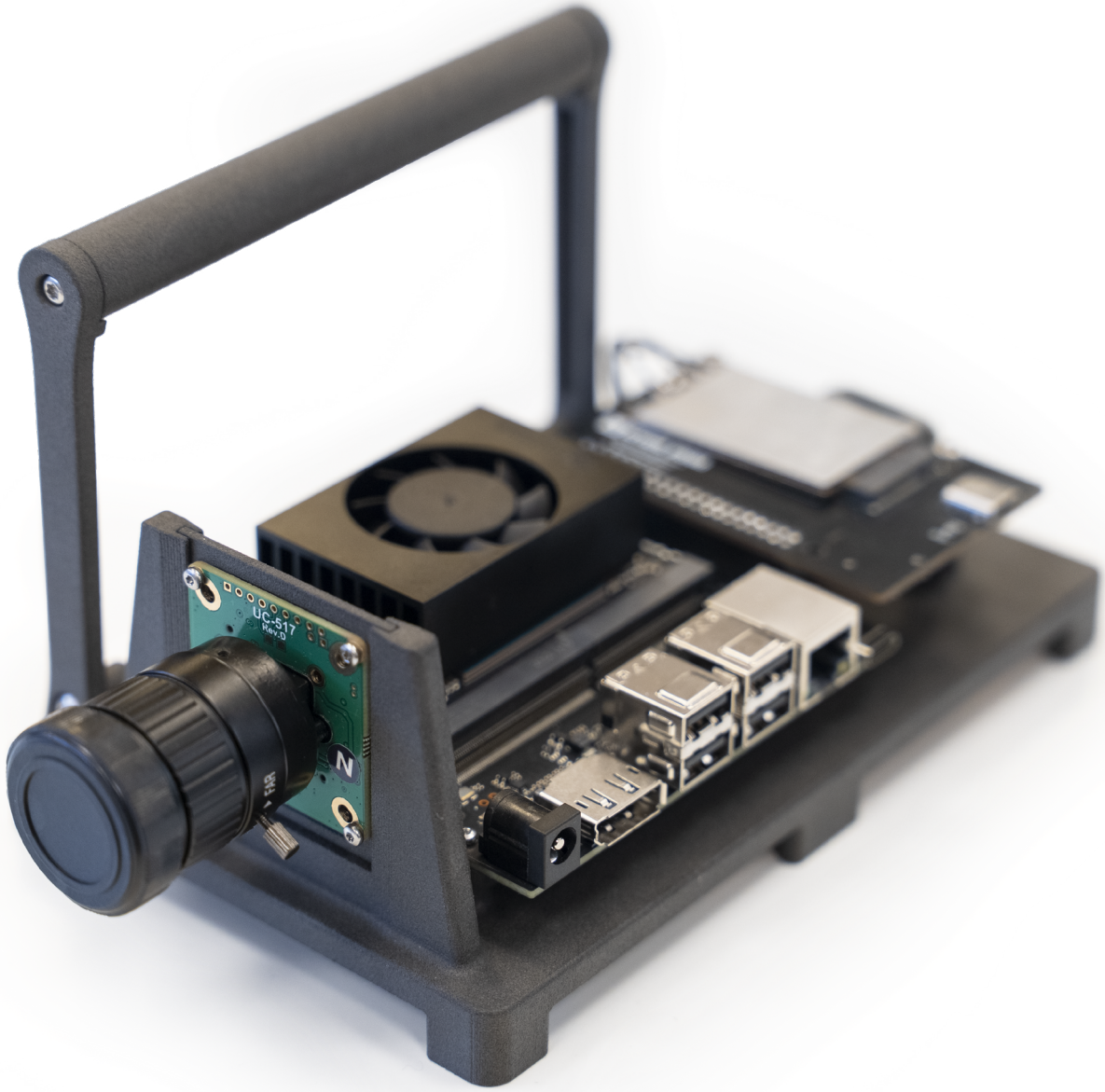
beCam

user's manual



BEYOND VISION

Contents

1 | Welcome

Thank you for purchasing Beyond Vision's **beCam**! This guide will help you started and profit the most from your most recent aquisition!

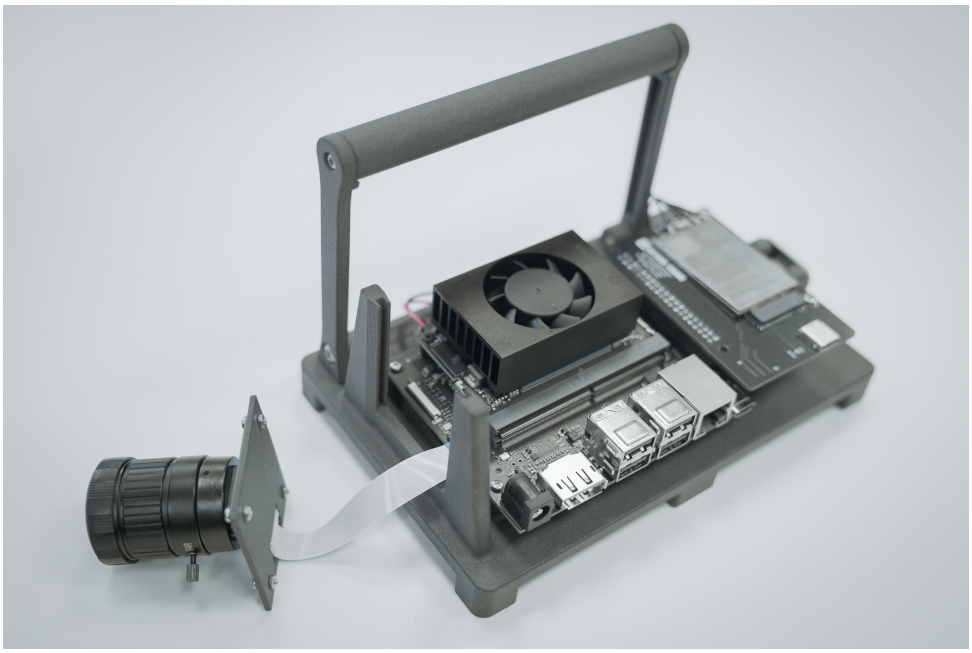


Figure 1.1: The beCam

Before you start, please take a moment to get to know the beCam: You can confirm the product and model by looking at the label (on bottom of the beCam) which must look like this one:

Figure 1.2: beCam's identification label

The beCam is composed of a Jetson Orin Nano, a Arducam B0249 and a 5G module and its carrier board, the system is capable of AI processing.

Among the many useful features of the beCam, the following stand out from the crowd:

- **Real-Time AI Detection and Tracking**: This kit performs **live person detection and movement tracking** directly on the Jetson Orin Nano. Students can walk in front of the camera and see bounding boxes instantly appear around them, with tracking numbers following their motion. This "immediate feedback" effect is powerful in a classroom because it transforms abstract AI concepts into something students can see and interact with, making AI fun, engaging, and easy to understand even for beginners.
- **Powered by NVIDIA Jetson Orin Nano**: At the heart of the kit is the Jetson Orin Nano, a compact AI computer capable of running advanced neural networks with surprising speed. This gives students access to the same technology used in robotics, autonomous systems, and

smart cities—all inside a classroom. Teachers don't need special hardware knowledge; the Jetson runs preconfigured software, creating a **professional, industry-grade experience** in an educational context.

- **DeepStream AI Pipeline (Simplified for Education):** DeepStream allows complex AI pipelines—camera input, preprocessing, inference, tracking, and visualization—to run smoothly in a single framework. What makes it stand out in this kit is how it is **configured to work out-of-the-box**, so students immediately see results while advanced learners can explore configuration files, tuning, and pipeline customization. This gives a smooth entry point for complete beginners and a deep technical playground for more advanced students.
- **Fully Integrated 5G Connectivity:** Unlike traditional classroom AI kits that require Wi-Fi or Ethernet, this kit includes a **5G module** for fast mobile connectivity. Students can send AI results to a server, stream video remotely, or simulate real-world robotics and IoT applications. This feature uniquely prepares students for the future of connected AI systems—drones, delivery robots, remote cameras, and autonomous sensors—all of which rely heavily on 5G networks.
- **Hands-On Learning with Real Hardware:** Many AI courses rely only on simulations or pre-recorded datasets. This kit allows students to work with **physical hardware**, real cameras, and real-time interactions. They see how lighting, distance, motion, and angles affect AI performance. This direct connection to the physical world gives students a deeper understanding of how AI actually behaves outside of ideal lab conditions—an essential insight for future engineers and scientists.

Please follow closely the setup and configuration procedures (Section 4), as well as the safety information (Section 3) to benefit the most from your beCam.

If you have any doubt or need more information on beCam's features or performance, please contact us by using the contact information below:

BEYOND VISION

Rua da Confersan, 2615-153 Alverca do Ribatejo, Lisboa PORTUGAL

+351 218 873 530 • +351 911 890 642

info@beyond-vision.com (commercial)

support@beyond-vision.com (technical support)

https://beyond-vision.com/

You are also invited to follow us on our social networks, by using the links below:

https://www.linkedin.com/company/beyondvision-join-the-future/

https://www.instagram.com/beyondvisiontech/

https://www.facebook.com/beyondvision.tech

https://x.com/beyondvision_pt

https://www.youtube.com/@beyond-vision

To acess the most updated information or latest news, Beyond Vision invites you to visit: docs Beyond-Vision where information is permanently being updated.

2 | Product Description

The beCam Developer Kit is an **open, skeleton-style** platform. Instead of hiding everything inside a closed box, we expose most components so that you can see how they look and how they connect to each other.

At the front there is a sliding door that allows you to split the camera from the rest in case you need to adjust the direction it is pointing. The rest of the components (Jetson Orin Nano, 5G module, power supply and cables) are placed on the base skeleton structure, clearly visible and easy to access. The power button is located on the opposite side of the camera which you can use to power the device on or off [1].

This open design is intentional. It helps you:

- Visualize the building blocks of a computer vision system with 5G connectivity;
- Understand how power, data and signals flow between components;
- Learn to handle hardware responsibly and carefully.

In Addiction, there is a handle so that you can carry the kit safely between rooms or tables in the classroom when needed.

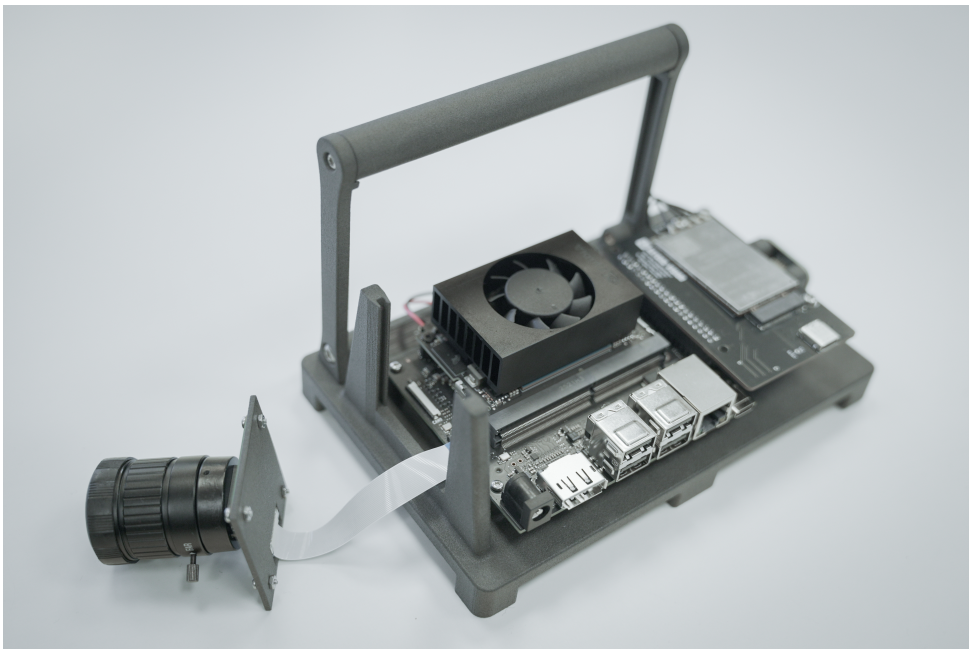


Figure 2.1: Side view

2.1 | What's in the box

When you open the box for the first time, you should find:

- The **beCam Developer Kit** assembled on its base structure
- The **IMX477 camera (Arducam B249)**[**?**] mounted behind the front sliding wall
- The **Jetson Orin Nano Developer Kit** installed on the skeleton base
- A **5G module** attached to the system (with its antennas)

[1] keep in mind that if you are connecting the power cable to the jetson you don't need to click on the power button (as soon as it receives power it will start, the button only works in case of manual shut down, sleep etc)

- **Power adapter and cable**
- **USB-A to USB-C cable** for 5G module to Jetson Communication

If something is missing or appears damaged, please inform your teacher or lab supervisor before powering on the kit.

2.2 | beCam Developer Kit

The beCam is the central platform of this product. It is a small AI computer built around the NVIDIA Jetson Orin Nano.

The beCam is organised to be visually clear:

- The camera is isolated at the front so you can easily see the field of view.
- The Jetson board is in the middle, acting as the "brain" of the system.
- The 5G module and antennas are placed to the side for better signal and easier identification.
- Cables are routed in a way that lets you follow the data path (camera Jetson network).
- Power button is behind.

You will use this kit to run AI pipelines that can detect people or objects, draw boxes around them, and track them over time.

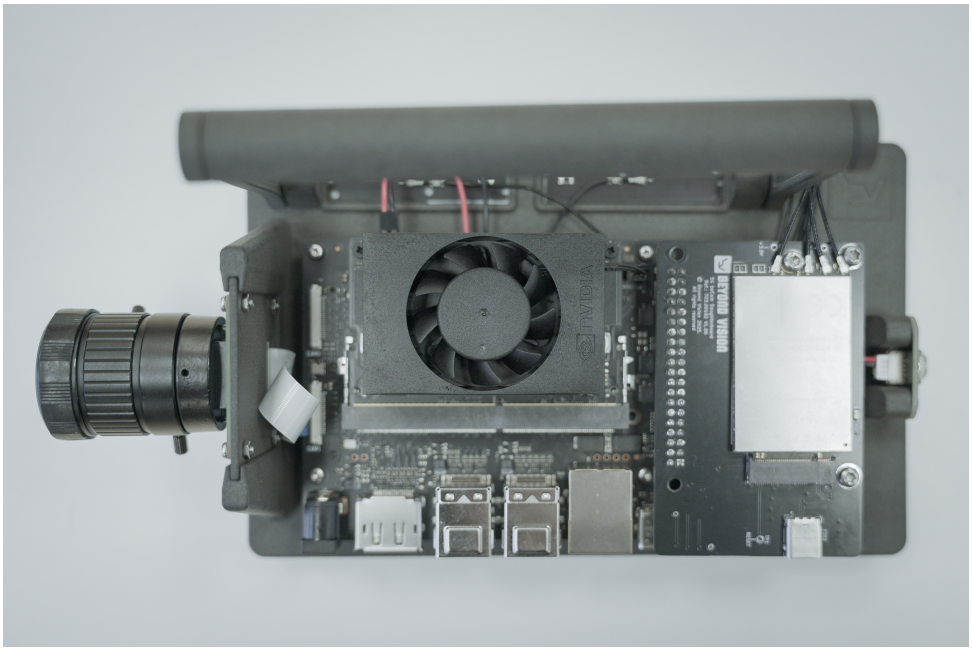


Figure 2.2: Top-down view

2.3 | Meet the Components

This section introduces the main hardware building blocks of the kit and explains the role of each one in the overall system. Figure 2.3 provides a visual reference of the assembled setup, while the following subsections describe in more detail the Jetson Orin Nano (computing and AI processing platform), the camera – Arducam B0249 (image acquisition sensor), and the 5G module (cellular communications and remote connectivity). Together, these components form the complete pipeline from capturing video, to processing it on-board, and finally enabling network access for monitoring and data transfer.

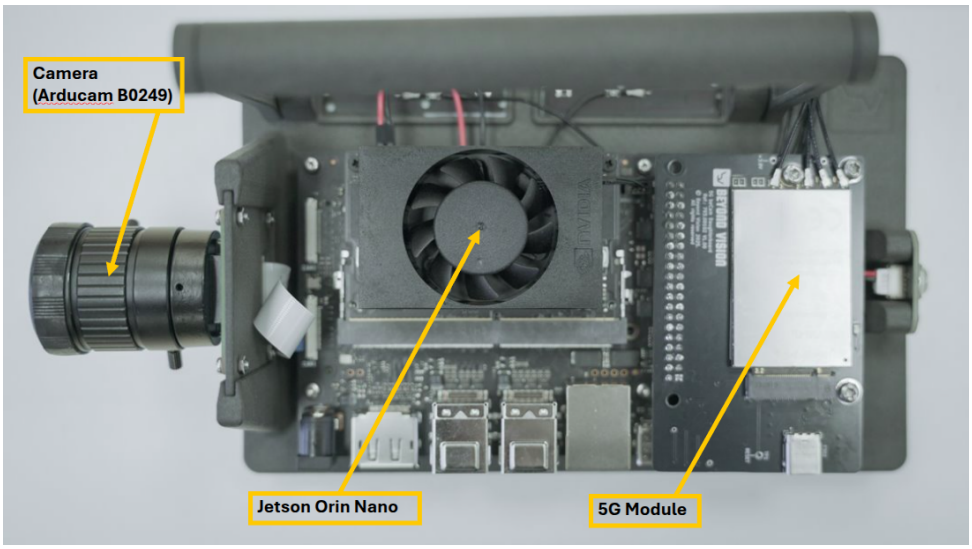


Figure 2.3: System overview showing the main hardware modules: Jetson Orin Nano, Arducam B0249 camera, and 5G module.

2.3.1 | Jetson Orin Nano

Jetson Orin Nano Developer Kit This small board is a powerful AI computer. It has:

- A **GPU** for accelerating AI models
- A **CPU** to run the operating system and applications
- Ports for camera, Ethernet, USB, HDMI and more

All processing for detection and tracking happens here using NVIDIA DeepStream (see Section 5.1).

2.3.2 | Camera - ARDUCAM B0249

IMX477 Camera (Arducam B249)[?] This is a high-quality camera module that is used to capture live video. It connects to the Jetson using a flat ribbon cable. You can adjust its position with the sliding door and gently rotating or repositioning the camera.

2.3.3 | 5G Module

5G Module The 5G module provides mobile connectivity. It can be used to:

- Send processed data to remote servers
- Connect the kit to cloud applications or dashboards
- Simulate real-world IoT and edge AI scenarios

3 | Safety & Responsible Use

Because the electronics are exposed and accessible, it is very important to use the kit carefully. This section explains the main risks and how to stay safe.
Always read this section before using the beCam for the first time.

3.1 | General risks concerning the usage

The beCam kit works at low voltages, but it is still an electronic device and must be treated with respect.
Possible risks include:

- **Electric shock or damage** if connectors are pulled, shorted or exposed to liquids
- **Hot surfaces** on the Jetson heatsink or power components after long use
- **Mechanical damage** if the beCam is dropped, carried by the cables or used as a support object
- **Privacy issues** if the camera is used to film people without their consent

You should never open power supplies, cut cables or modify the electronics unless instructed and supervised by your teacher.

3.2 | General safety guidelines

To use the beCam safely:

- Place it on a **stable, flat surface** with enough free space around it.
- Do not operate it with **wet hands** or near liquids.
- Do not cover the Jetson heatsink or block airflow; it needs to "breathe" to stay cool, especially when running AI Models.
- When carrying it, always use the **handle** and support the base with your other hand if needed.
- Avoid touching connectors and pins when the system is powered on.
- Treat all cables gently. Do not bend them sharply, pull them by the wire, or step on them.
- Use the camera responsibly. Respect rules and your school's policies about filming people.

If at any point you see smoke, smell burning, or hear unusual noises, **unplug the power immediately** and inform your teacher.

3.3 | Maintenance safety guidelines

From time to time, the kit may need light cleaning, cable checks or adjustments to the camera position.
When performing any maintenance:

- Always **power off** the kit and unplug it from the mains first.
- Wait a few minutes for components to cool before touching the heatsink.
- Use a dry, soft cloth to clean dust from the base or the camera housing. Do not use liquids.
- If a cable seems loose, gently push the connector, never force it at an angle.
- Do not open or modify the power adapter.

If something looks broken or you are unsure how to fix it, **do not experiment alone**. Ask your teacher or lab supervisor.

A very important point concerns the flat ribbon cable that connects the camera to the Jetson:

- If you ever need to **change, reinsert or adjust the flat ribbon cable, you must do it only with the kit completely powered off and unplugged.**
- Never pull the cable by the film itself; gently unlock the connector, adjust the cable, then lock it again.
- It's position is shown on the Figure 3.1.

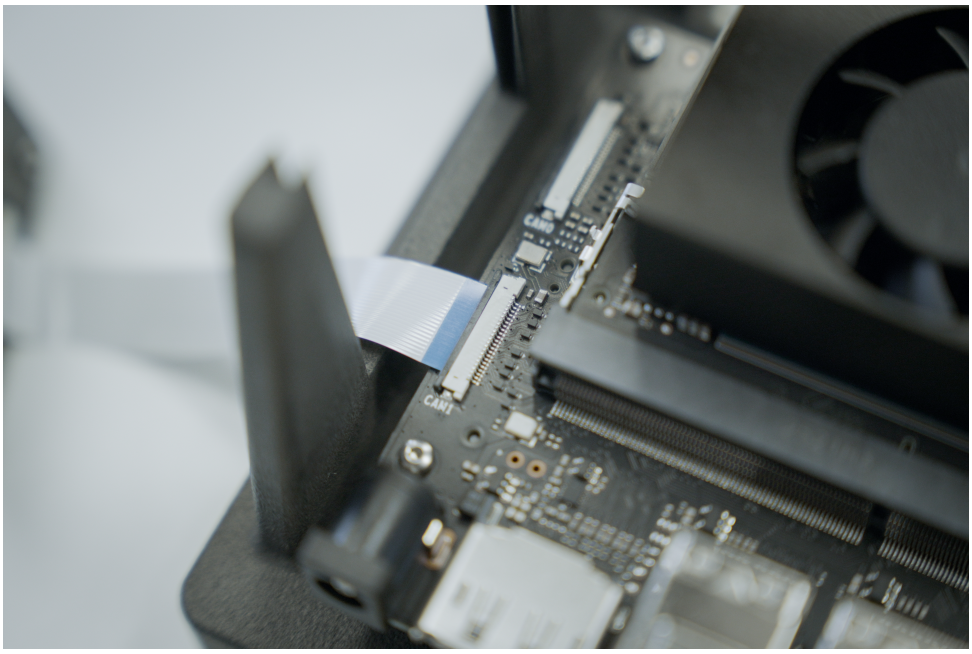


Figure 3.1: Close-up of the camera flat ribbon cable showing the correct position and connector lock

Another important aspect of maintenance is **software updates**:

- The kit is delivered with **JetPack 6.2.1** and **DeepStream 7.1** pre-installed and validated.
- You should **not update JetPack, the operating system or DeepStream** without confirming all compatibilities.
- Before any update, it is necessary to check:
  - The **DeepStream version vs JetPack version** compatibility on the official NVIDIA documentation.
  - The **camera driver compatibility** for the target JetPack version on the **Arducam** website.

If you are not sure, do not run updates on your own. Software changes should only be made by a teacher or system administrator who has verified these compatibility matrices.

4 | Installation and Configuration

This section explains how to set up the beCam from a powered-off state until it is ready to run the demo.

4.1 | Connecting Everything

1. **Position the kit** - Place the beCam on a stable table, with the camera facing the area you want to monitor.
2. **Adjust the camera** - If needed, slide open the front door and gently adjust the camera's direction. Make sure the ribbon cable is not twisted or under tension. You can also manually adjust light and focus.
3. **Connect display and peripherals (if used)**:
   - Connect an Display-Port cable from the Jetson to a monitor.
   - Connect a USB keyboard and mouse to the Jetson's USB ports.

   In some setups, you may access the Jetson over the network using SSH instead of using a local monitor. Follow your teacher's instructions.
4. **Connect to network or 5G**
   - If you are using Ethernet, plug the Ethernet cable into the Jetson and the school network switch.
   - If you are using Wi-fi, follow your school Wi-Fi Set-up.
   - If you are using 5G, there are two things to check:
     - Make sure the 5G antennas are firmly attached to the module.
     - Connect the USB-A to USB-C cable between the Jetson and the 5G module.

     This USB connection is what allows the Jetson to receive internet access from the 5G modem. If this cable is not connected, the module may power on, but the Jetson will not have 5G connectivity.
5. **Connect power** - Connect the power adapter to the Jetson's power input and then to a power socket.

4.2 | Powering On

To power on the kit:

1. Make sure all cables are connected properly.
2. Press the power button on the Jetson Orin Nano (if present on your kit) or simply plug in the power adapter, depending on the configuration.
3. Look for status LEDs on the Jetson board next to the USB-C. A steady LED usually indicates that it is booting.
4. After a short time, you should see the boot messages or login screen on the monitor.

If nothing appears on the screen after a few minutes, refer to the Troubleshooting section (Section 8).

4.3 | Basic Connectivity

Once the system has booted:

- If you are using a direct monitor, you will see the desktop environment or a login prompt on the screen.
- If you are connecting over the network, your teacher will tell you the IP address or hostname of the Jetson. You can then connect using SSH or a web-based interface, depending on the lab instructions.

In general, you will:

1. Log in with the username and password provided by your teacher.
2. Verify that the system can access the network (for example, by opening a web browser or using a simple network test).

You can record your credentials in the Appendix so you do not forget them.

4.4 | beCam setup

The beCam usually comes with a prepared environment for you, including:

- DeepStream installed and configured
- The PeopleNet model downloaded and ready
- One example pipeline
- And other examples on DeepStream.

If it is your first time you can start by:

1. Open a terminal and go to the `becam2` directory.
2. Once there you will run a command *python3 main.py -c config.bext*, in case of our demo you can find more instructions and a deeper explanation in Appendix A.3.
3. Waiting a few seconds for the pipeline to start.

If you are supposed to see the output in the Beyond-Vision software beXStream®, the exact steps will be explained in Section 6.
Once running, you should see (under the menu Monitor, Streams, selecting the camera) the camera image with boxes around detected people.

5 | Software Overview

This section gives you a basic understanding of the software inside the beCam. You will work with these tools in more depth during the course, but it is useful to know the main ideas from the start.

5.1 | Deepstream

DeepStream[**?**] is a software platform created by NVIDIA that allows computers like the Jetson Orin Nano to analyze live video using artificial intelligence. Think of it as a set of tools that helps the camera, the AI model, the tracker, and the display all work together smoothly. Normally, processing video frame-by-frame with AI would be too slow, but DeepStream is designed to make everything extremely fast — even fast enough to detect and track people in real time, without needing a big computer or a cloud server.

DeepStream [**?**] works using something called a **pipeline**. A pipeline is like an assembly line in a factory: video comes in from the camera, gets converted to the right format, goes through the AI model that finds objects, passes through the tracker that assigns IDs, and then finally displays the results on screen or sends them over the network. Each step in the pipeline can be customized, allowing students to change colors, thresholds, resolutions, or even swap the AI model for another one. Because DeepStream is built on top of GStreamer (a multimedia framework), it is flexible, efficient, and perfect for educational experiments.
To learn more, NVIDIA provides excellent official resources. Here are some of the most useful links:

- **DeepStream Official User Guide:**[2]
  https://docs.nvidia.com/metropolis/deepstream/dev-guide/
- **Jetson Community Projects (Great for classroom inspiration):**
  https://developer.nvidia.com/embedded/community/jetson-projects?sortBy=jetson_community_projects%2Fsort%2Fdate_added%3Adesc
- **DeepStream GitHub Samples & Code:**
  https://github.com/NVIDIA-AI-IOT/deepstream_reference_apps

These resources are completely free and offer example projects, sample code, configuration files, and step-by-step tutorials.

5.2 | Running a Pre-Built Demo

Out of the box, the beCam kit includes a demo pipeline using the *PeopleNet* model. PeopleNet is an AI model trained to detect people in images and videos. In this demo, the camera sends live video to the Jetson, DeepStream passes the frames through PeopleNet, and the result is a video stream with bounding boxes around the detected people.
To start the demo, you will typically run a Python script from the `becam2` directory. It is very important to pay attention to the directory where you are working, so that you do not accidentally modify or run files in the wrong place.

1. Open a terminal on the Jetson Orin Nano.
2. Navigate to the `becam2` directory. For example:

   ```
   cd becam2
   ```

   You can use the command `ls` to list the contents of the current directory and verify that you are in the correct location.
3. Once you are inside the `becam2` directory, run the demo command:

[2] Once on the site please check on whitch version of Deepstream you are seeing the documentation, by default the site takes you to the most recent version but the Kit on the version 1 of the manual has the Deepstream 7.1


```
python3 main.py -c config.bext
```

After starting the script, the beCam pipeline will initialize. At this point, you can check the video stream in the beXStream® application. You should see the camera image, with rectangles drawn around each detected person as shown in the example in Figure 5.1.

Figure 5.1: Example

This demo comes with a pipeline that is already configured and working. You are encouraged to explore and change parameters later in the course, but it is a good practice to always keep a copy of the original configuration and files so that you can return to a known working state if something goes wrong.
When you start the demo and everything is correctly configured, you should see:

- A live video feed from the IMX477 camera.
- Bounding boxes drawn around people detected by PeopleNet.
- Stable, real-time processing (if the system is not overloaded).

In later exercises, you can use this same structure to test other models, adjust thresholds, or integrate the video and metadata into more complex applications.
When you start the demo:

- The camera sends frames to DeepStream.
- DeepStream passes each frame through PeopleNet.
- PeopleNet finds people and returns bounding boxes.
- DeepStream draws boxes around each detected person and may label them with IDs for tracking.

You will see:

- A live video feed from the camera
- Rectangles around people in the scene
- Optional labels or IDs that allow you to follow each person as they move

During your labs, you might be asked to:

- Move in front of the camera and observe how tracking behaves
- See what happens when several people enter the frame at the same time
- Check how far the camera can see and still detect people correctly

This demo is your **starting point**. From here, you can begin to explore how AI models interact with video streams.

5.3 | Try Changing Something or Creating Your Own Model

After you understand the basic demo, the next step is to experiment.
Some simple changes you might try (under guidance):

- Adjusting the confidence threshold, to decide when a detection is considered valid
- Changing how the boxes look (colour, thickness, labels)
- Limiting detection to a certain region of interest in the image (for example, only the lower half of the frame)

Later you may also:

- Replace PeopleNet with another pre-trained model (for example, for cars or everyday objects)
- Use your own dataset to train a custom model, then integrate it into the DeepStream pipeline

By the end, you should understand not only how to run AI demos, but also how to design and modify your own video analytics applications.

5.4 | Software Updates

Version compatibility (for teachers / advanced users)
The beCam is shipped with a tested combination of JetPack 6.2.1, DeepStream 7.1 and the corresponding camera drivers. Before upgrading any of these components, always verify the official compatibility information:

- **DeepStream vs JetPack compatibility [?] on the NVIDIA website.**
- **IMX477 / Arducam driver support [?] for the target JetPack[?] version on the Arducam website.**

Updates should only be performed by someone who has checked this information and is prepared to reconfigure the system if needed.

6 | beXStream® platform

beXStream®is the cloud companion for Beyond-Vision's products. For beCam2, works as a cloud-based video sharing relay and video storage service.
If you purchased a beXStream®platform license, you were given a login to acess the application either by using a web browser. You can register and access the application by using this login and acessing the web address: beXStream®.

6.1 | Setup

In the scenario that you have purchase a beXStream®platform license, you are entitled to acess the application. The beXStream®can be run using a web browser. Please follow the instructions below to successfully access the platform.

6.1.1 | Web Browser

The beXStream®platform runs on any of the major web browsers (and it has been validated mostly on Chrome and Firefox). To access the application, you should open your browser and type in the following address: beXStream®. You should see a screen like the one shown in the Figure 6.1 below:

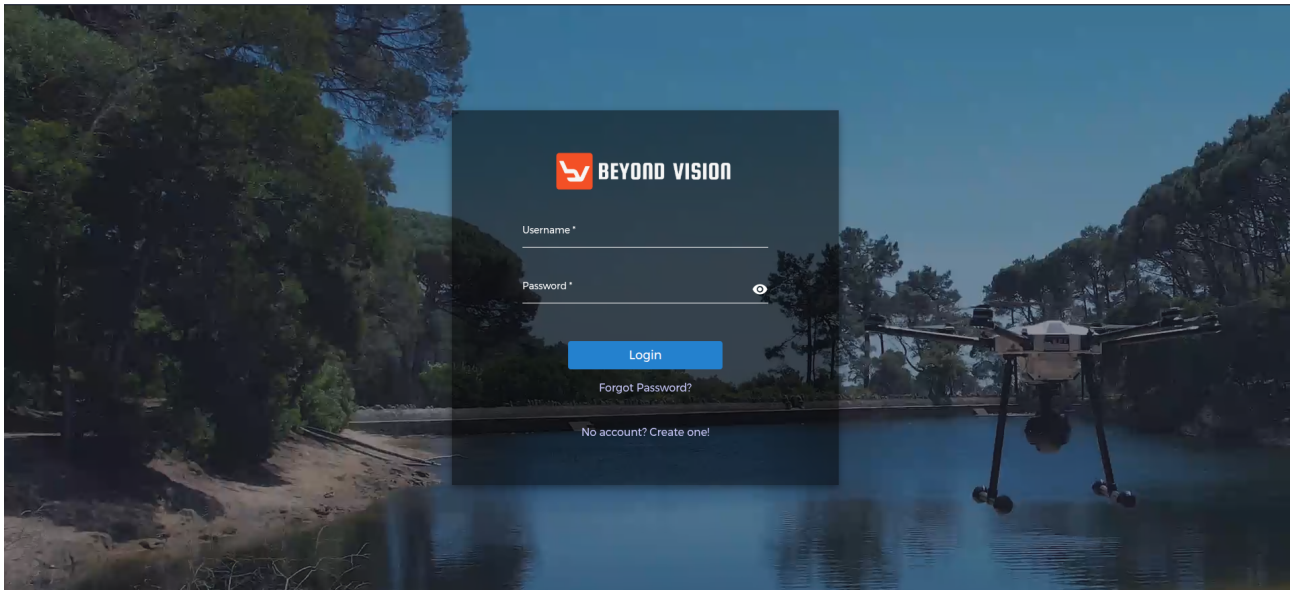


Figure 6.1: beXStream®web browser login screen.

Type in the username given in the login field, and the password, to authenticate and start using the application.

6.1.2 | Update Password and Email Address

After the first login, you should start by changing the password you were given to for a password of your choice.
You should click on the *Edit Profile* link as shown in fig. 6.2,

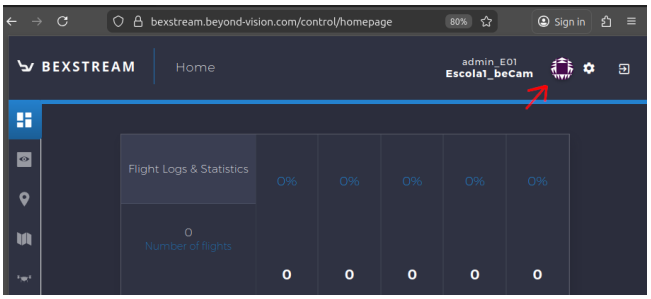


Figure 6.2: beXStream® Self-Edit profile.

and supply a new password. You should also supply an email address. The email address **is required** to make use of the Forgot password feature.

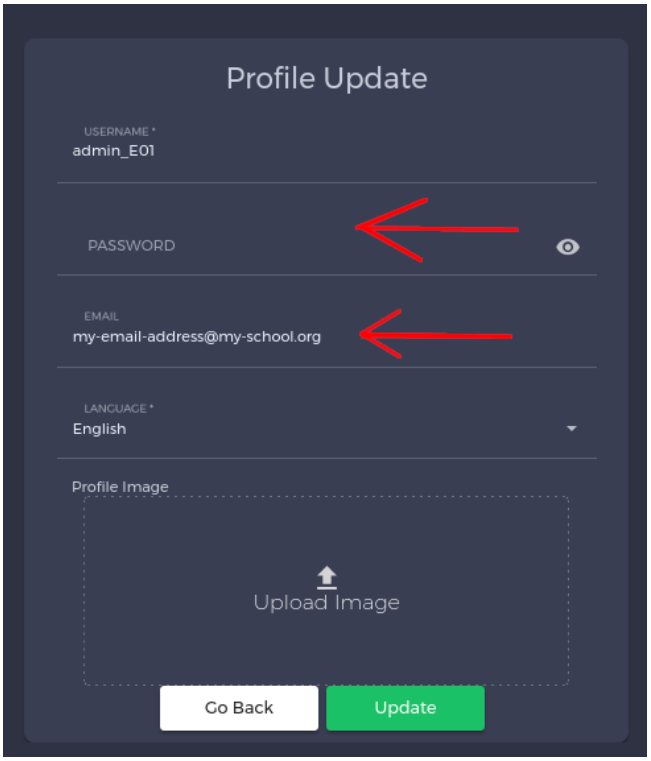


Figure 6.3: beXStream® User data.

Optionally, you can also provide a picture, or if you have a gravatar.com© avatar associated with your email, it will be visible in the upper right corner just besides the username (where the *Edit Profile* link is located, as shown in fig. 6.2).
Now, you can start exploring all the functionalities provided by the platform.

6.2 | Pre-Provisioned Users

Each organization that purchases a beCam2camera is pre-provisioned with a set of pre-defined users:

E01_admin - An administrator user, with profile **admin** that has full access to all features of the platform. **E01** is the organization identifier, and should be replaced by your own organization identifiable (E02, E03, E04, etc...).

estudante1_E01, estudante2_E01, estudante3_E01,... - Student users with profile **student** limited access to the platform, only able to view video streams and download recorded videos.

bv-support-E01 - A user to allow Beyond Visonto provide support to the organization. **This user should not be deleted**.

6.2.1 | User Management by the Organization's Administrator

In the Users menu, the organization's administrator **E*nn*_admin** with **admin** profile can perform user management: creation, editing, and deletion of existing users.

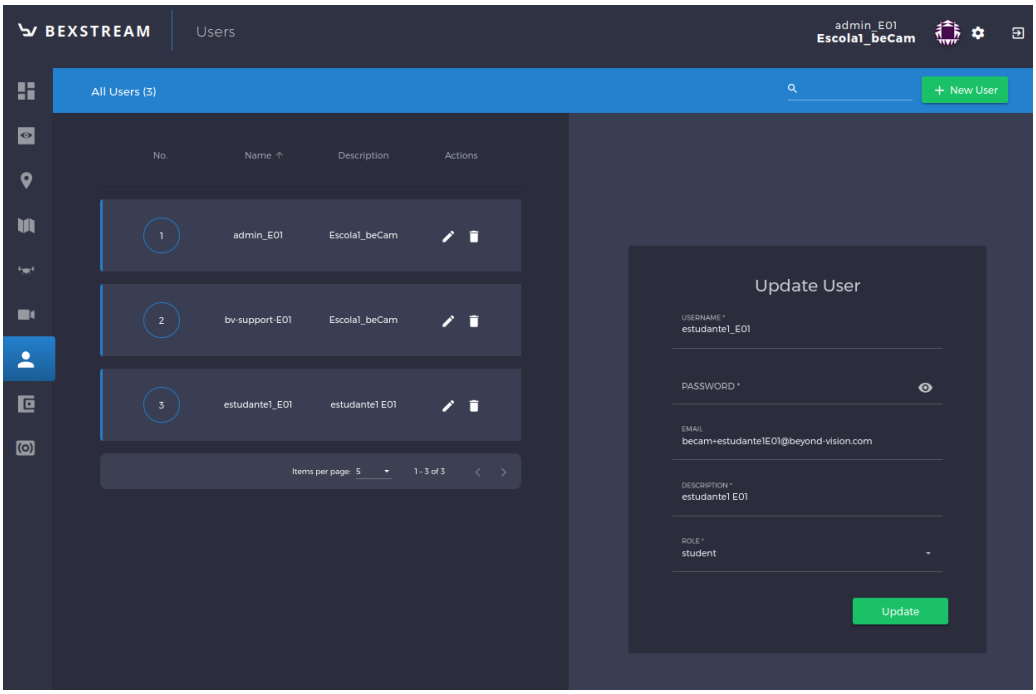


Figure 6.4: beXStream® User Management by the administrator.

To create a new user, press the **New User** button; fill in the desired properties for the new user; press the **Create** button.
To change one property of an existing user, press the edit icon next to the user; edit the properties that need to change ; press the **Update** button. For example, to **change only the password** of an existing user, press the edit icon next to the user; set the desired password (taking care not to edit any other attribute of the user); press the **Update** button.
Press the delete icon next to the user to delete an existing user.

6.2.2 | Student User Profile

The **student** user profile has limited access to the platform, being able only to view real-time video streams and download recorded videos from cameras associated with the organization, as illustrated by the main menu in fig. 6.5.

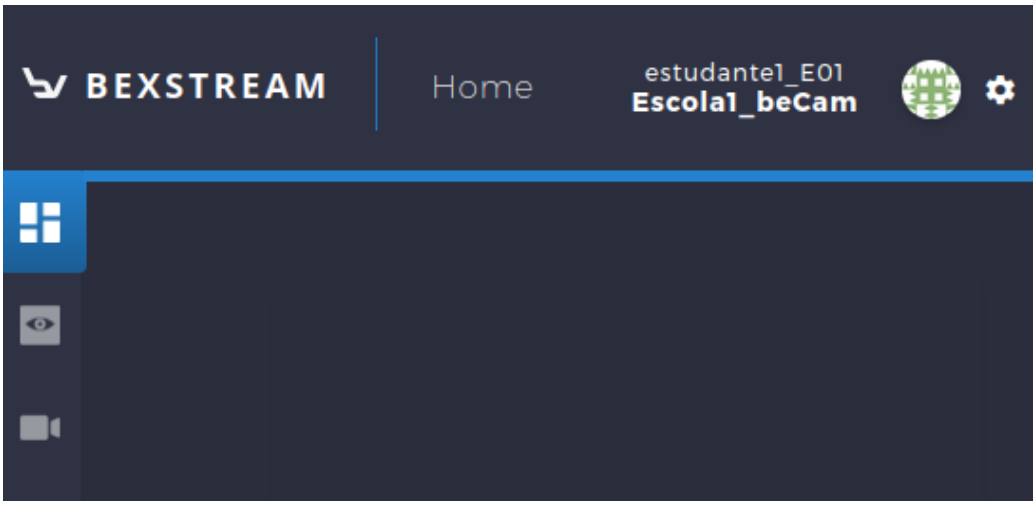


Figure 6.5: beXStream® Menu Options for Student Profile User's.

6.3 | Forgot password

If the password is forgotten or lost, it can be reset given the email address associated with the username.
Sending a password reset email can be requested on the login screen by clicking on the Forgot Password link (see fig. 6.6).

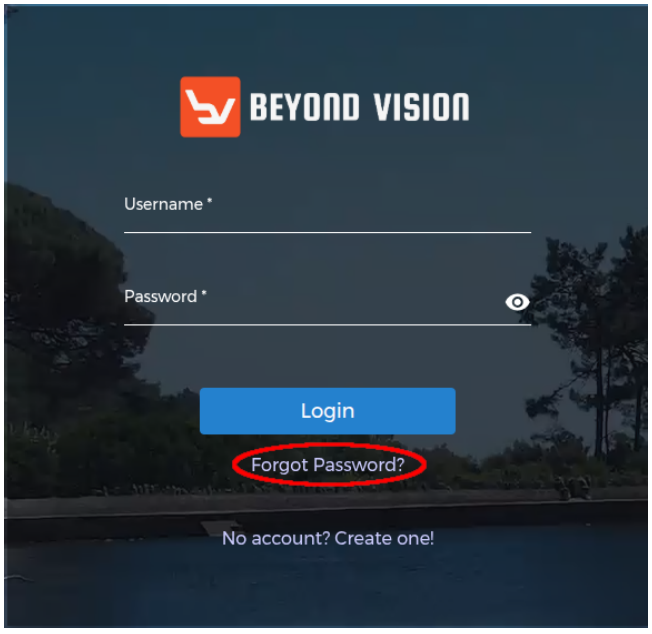


Figure 6.6: Forgot Password option on the login screen

Then, supply the email address associated with the username that the password needs to be reset (and pass the necessary tests to make sure you are a human user) (see fig. 6.7).

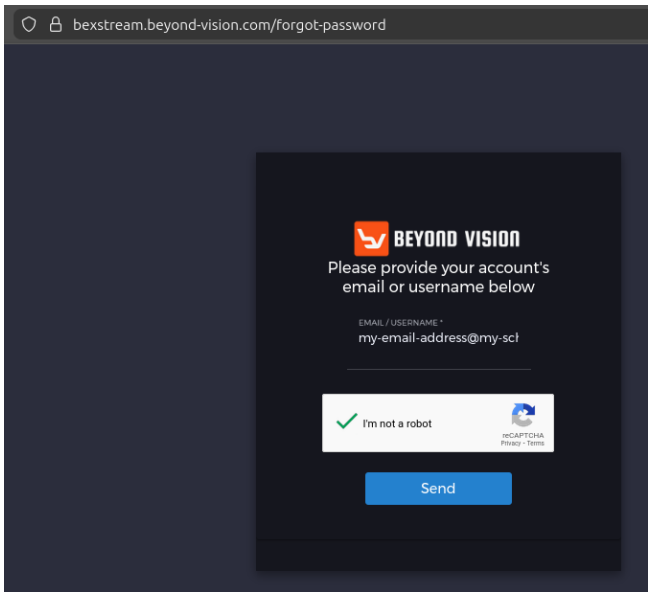


Figure 6.7: Forgot Password option on the login screen.

An email with password reset instructions will be sent to the email address (if that email address matches a user registered on beXStream®).

6.4 | Cameras

Listing existing cameras associated with your organization is done by selecting the Other Assets, Cameras menu option (see fig. 6.8).

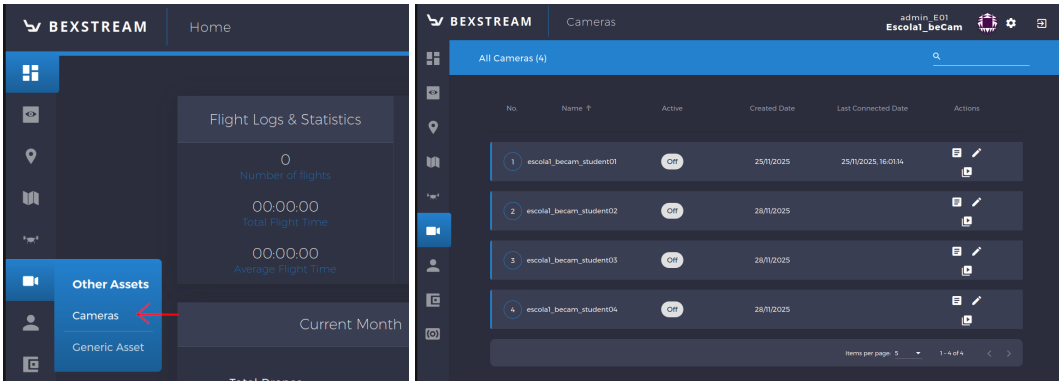


Figure 6.8: Cameras associated with the organization.

The list of cameras will be displayed.

6.4.1 | Camera status

If a camera is online and streaming video, the status will show “On” in green (see fig. 6.9).

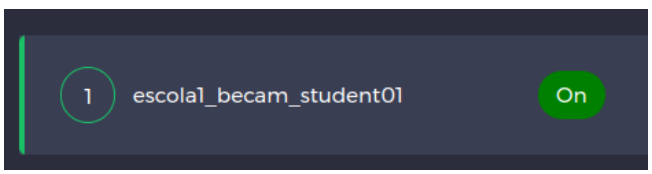


Figure 6.9: Example of an online camera status display.

For beCam2 cameras, status = On is also an indication that the camera is streaming a live video stream that is being uploaded and recorded in beXStream®.

6.4.2 | Downloading camera configuration file

The set of beCam2cameras supplied by Beyond Visonare pre-configured for each organization, so this feature will only be needed if the camera is reset to factory settings.
The camera configuration file can be downloaded by clicking on the icon. This will download a binary file named "config.bext" that is already installed in the beCam2box contains the necessary information for the camera to connect (by internet) to beXStream®. It is unique for each camera asset. (Should the same configuration file be installed in two different cameras, unexpected behavior may occur).

6.4.3 | Camera name

The edit icon allows changing the camera "NAME" (see fig. 6.10).

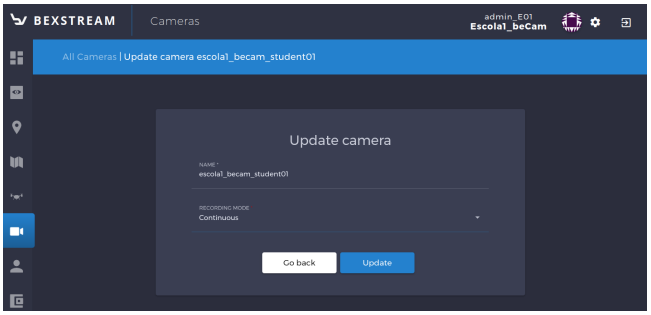


Figure 6.10: Editing camera name.

For beCam2 cameras, the "RECORDING MODE" is ignored.

6.4.4 | Browsing recorded videos for a particular day

The edit icon allows browsing the videos recorded by the camera for one particular day[3] (see fig. 6.11).

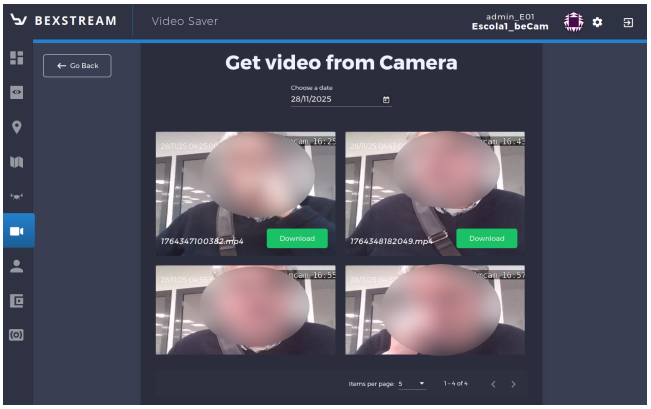


Figure 6.11: Browsing videos by date.

Each video can be downloaded as an .mp4 file by clicking on the "Download" button.
The calendar date picker ill show the days that have videos recorded shaded in a grey circle, and the current picked day as a green circle (see fig. 6.13)..

[3]The first time that this feature, new videos will be shown as simple square symbol. On further access, it will animate a few of the initial frames of each video on mouse-over.

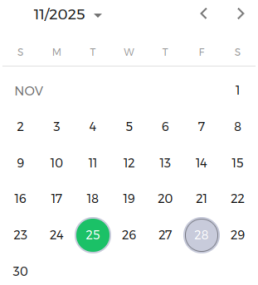


Figure 6.12: Days with videos are indicated by a grey circle.

In the current version of beXStream®, only one day at a time can be browsed.

6.5 | Realtime video streams

In the menu Monitor, Streams, you can view all current active video streams from all active cameras.

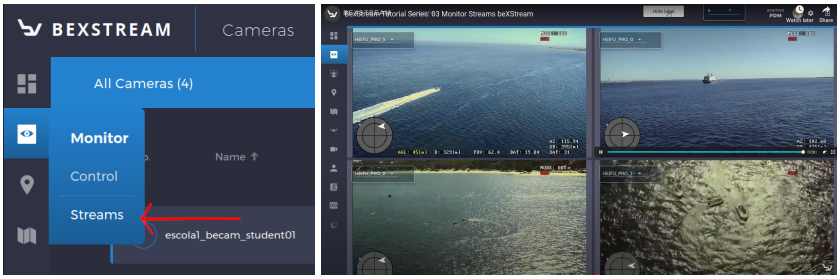


Figure 6.13: Streams show several real-time video feeds on-screen at the same time.

For a live example on how to use this feature, please refer to the Monitor, Streams Tutorial on YouTube, or look into the homepage Beyond Vison and look for Products, beXStream®, Videos.

6.6 | Re-configuring after Factory Reset

If the board is reset to the factory image (see Section 7), association with the beXStream®camera asset must be performed manually.
Presuming that you have already set the hostname to the original hostname (based on the serial number), use the administrator login on to beXStream®, access the Cameras menu, and download the camera configuration file (6.4.2). Install that file in ... TODO.

6.7 | General Data Protection Regulation compliance

The beXStream®platform does not ask for personally identifiable information:

- The pre-provisioned usernames are associated with organizations that have acquired a Beyond Visonproduct, and contains no personally identifiable information (is not relatable to the name of an actual person, but to a professional organization role).
- It is recommended that the email address given (in sec. 6.1.2) to be a professional one (instead of a personal one). In the case a personal email address is given, the user (or the organization's administrator) can change it at any time.
- The data captured by Beyond Vison's devices (videos captured from cameras/drones/etc) and uploaded to beXStream®is stored under access control rules that limit the access to such content to users associated with the same organization that owns those devices, and as such, are sole responsibility of that organization.

- Information related to the usage of the platform (login times, IP addresses, etc) is stored for security and auditing purposes (for a limited period of time), and is not shared with any third party.

6.8 | beXStream version

This manual was written based on beXStream®'s frontend version 1.5.6 on 21.01.2026. Future improvements and changes to the platform may make some of the screenshots and descriptions in this manual slightly outdated (in some details), but the core functionality described here will be preserved.

7 | Systems Image Recovery

This subsection is intended for professors and lab administrators who are responsible for maintaining the beCam Developer Kits.

When to consider a full recovery A full system image recovery is recommended when:

- The Jetson no longer boots into the desktop or console.
- DeepStream or the beCam demo fails in a way that is not fixed by simply restoring configuration files or the original pipeline no longer works.
- Someone has attempted a major update (for example, a JetPack or DeepStream upgrade) without checking compatibility and the system becomes unstable.

Whenever possible, copy any important student work off the device (for example, via USB drive or network share).

IMPORTANT: If a student breaks the software environment, please contact Beyond-Vision to do a full image recovery.

8 | Troubleshooting

Even with a prepared kit, issues can happen.This section provides solutions to the most common and known issues encountered during installation, configuration, or execution of the system. Problems are organized by category, with symptoms, causes, and corrective actions.

Symptom: The kit does not power on

- Check that the power adapter is firmly connected to both the socket and the Jetson.
- Try a different power socket or power strip.
- Look for any LEDs on the Jetson board. If none are on, inform your teacher – the power supply or board may need technical inspection.

Symptom: The screen stays black

- Make sure the monitor is powered on and set to the correct Display Port input.
- Check the Display Port cable at both ends.
- If possible, try another Display Port cable or monitor.
- If still no image, power cycle the kit and watch for any change in LEDs or messages.

Symptom: No camera image in the demo

- Confirm that the demo pipeline is actually running (no error messages in the terminal or interface).
- Check the camera cable. It should be fully inserted and not loose.
- Make sure nothing is blocking the camera lens (the light regulator could be completely close).
- If the system shows an error related to the camera, inform your teacher; the camera may need to be re-seated or replaced.

Symptom: No 5G connection

If you expect the Jetson to be online via 5G but there is no connectivity:

1. **Check the 5G module LED**
   - Is the LED on the 5G module powered on?
   - If the LED is off, the module may not be powered or not correctly connected.
2. **Check the USB-A ⟷ USB-C cable**
   - Ensure the **USB-A to USB-C** cable is connected between the Jetson and the 5G module.
   - Without this cable, the 5G modem cannot provide internet access to the Jetson.
3. **Check the SIM card**
   - Make sure a **SIM card** is inserted in the module's SIM card slot.
   - Confirm that the SIM is active and has a valid data plan.
4. **Check if the SIM has a PIN**
   - If the SIM uses a **PIN code**, the module will usually not connect to the network until the PIN is handled.
   - There are two main options (normally for teachers or advanced users):
     - Remove the PIN using a phone or another device.
     - Or send the appropriate **AT commands** to the module to unlock the SIM and configure the connection.

If you are not comfortable with AT commands, do not attempt this alone; ask your teacher or lab supervisor.

Symptom: The demo runs, but there are no boxes around people

- Stand closer to the camera and ensure you are fully inside the frame.
- Make sure the lighting is reasonable (not completely dark or with very strong backlight).
- Ask your teacher to check if the PeopleNet model and configuration are correctly loaded.
- In some cases, thresholds may be set too high; this can be adjusted in configuration files by the instructor.

Symptom: Pipeline error "Failed in mem copy"

If you see an error similar to **"Failed in mem copy"** when running a DeepStream pipeline, it may be related to how memory copy is handled by a specific element in the pipeline.

A common workaround is to **set the property** $fcopy\_hw = 2$ on the relevant element in the DeepStream configuration file.
For example, in the configuration file used for the pipeline, within the section of the element that is failing, you would add a line such as:

```
fcopy_hw=2
```

This change should be done by your teacher or an advanced user who manages the configuration files. After updating the config, restart the pipeline and check if the error disappears.

Symptom: Bad image quality (blurry, too dark or too bright)

The IMX477 camera in this kit has manual focus and manual light/exposure control. It does not behave like a phone camera that automatically focuses and adjusts exposure perfectly in every situation.
If the image looks bad:

- If it is **blurry**, ask your teacher before carefully adjusting the **focus ring** on the lens while watching the image on the screen until it looks sharp.
- If it is **too dark or too bright**, ask your teacher before carefully adjusting the **light ring** on the lens while watching the image on the screen until it looks sharp.
- In more advanced setups, the teacher may adjust camera exposure parameters in software.

If a problem persists after these checks, document what you see (error messages, LED states, what you did before it failed) and share that information with your teacher. Clear information helps solve issues faster.

beCam® User's Manual, version 1.01, 21.01.2026.

A | beCam Developer Kit Appendix

A.1 | Ports and Connectors Map

This section provides an overview of the main ports and connectors used in the beCam. Students can refer to these diagrams and photos when connecting the kit or during exercises.
Figure A.1 illustrates the top view of the ports on the beCam.

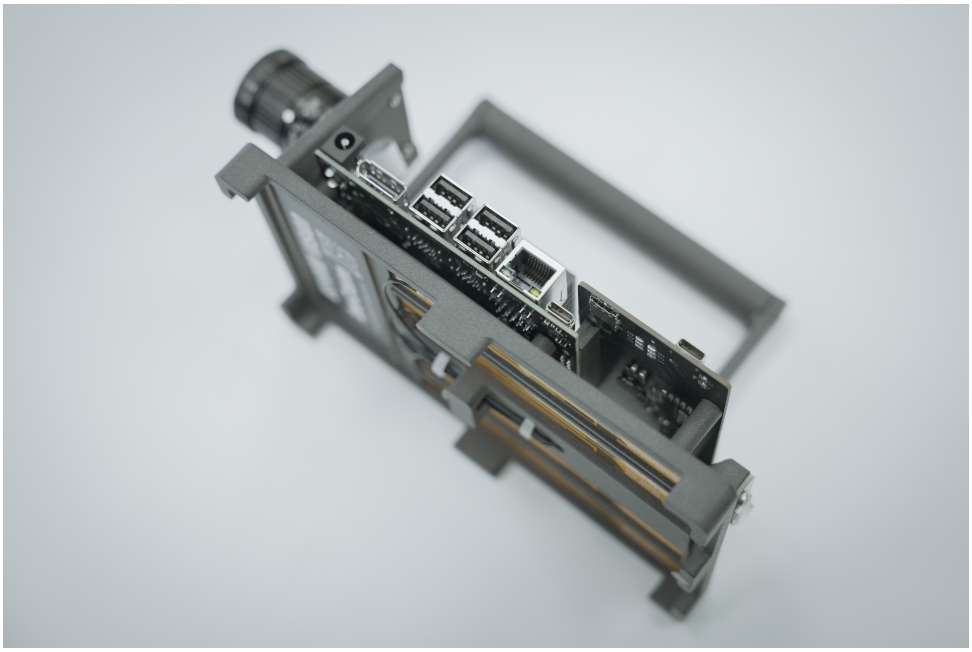


Figure A.1: Top view of the beCam Developer Kit with labeled components and connectors.

A more detailed view of the Jetson ports is shown in Figure A.2. This can be useful when identifying specific interfaces such as Display Port, USB, Ethernet and power.

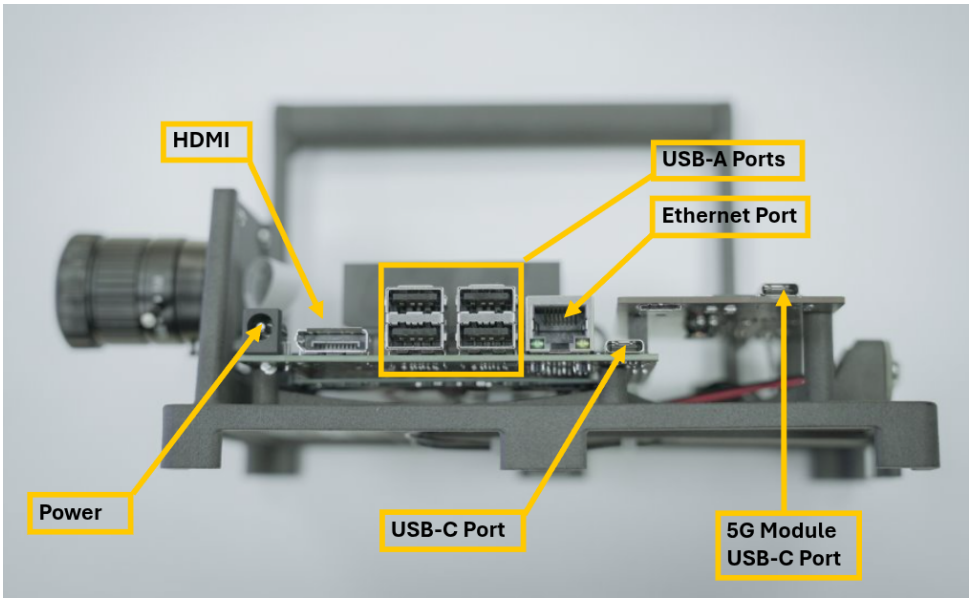


Figure A.2: Close-up of the Jetson Orin Nano ports with labels.

NOTE: Keep in mind that the USB-C Port is primarily a USB 3.2 data port used to connect the Jetson to another computer for setup/debug and flashing.

Finally, Figure A.3 shows the flat ribbon cable that connects the camera to the Jetson. The photo highlights the correct orientation of the contacts and the locking tab.

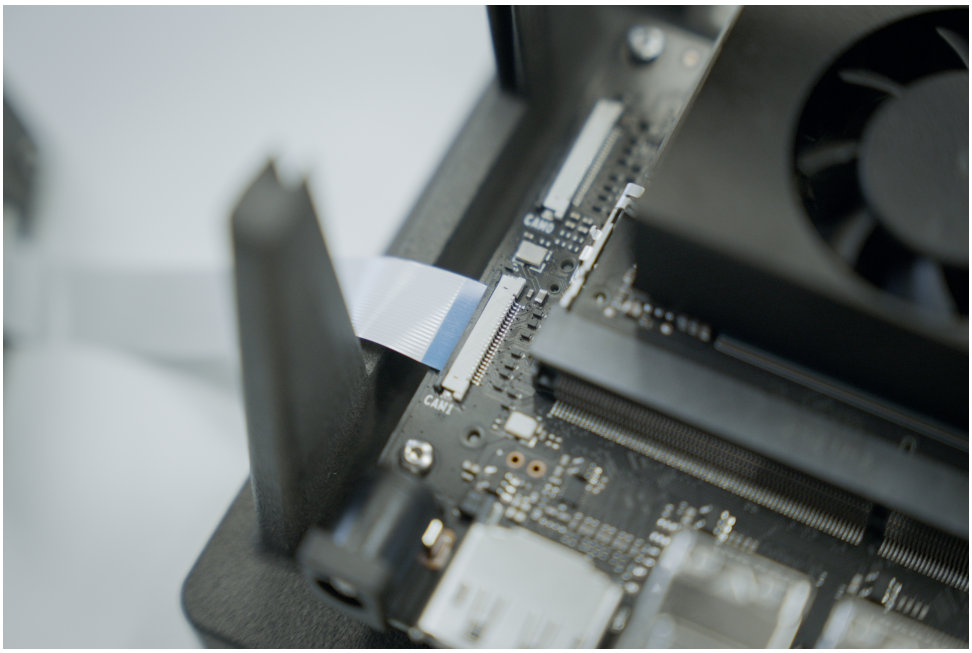


Figure A.3: Camera ribbon cable and connector. Adjust or replace this cable only with the kit powered off and unplugged.

A.2 | User Notes and Credentials

In many laboratory environments, the beCam Developer Kit is used with specific user accounts and network settings defined by the school. This subsection provides a simple place to record those details so that students do not lose access during exercises.

The table below can be printed and filled in by hand or completed digitally before distributing the manual to students.

Username	becam2
Password	becam
IP / Hostname	
Demo command or launcher	

Table A.1: Space for recording user credentials and access information for the beCam Developer Kit.

Students should keep this information private and follow the school's security guidelines. Passwords should not be shared outside the class or lab group.

There may also be additional notes useful for a particular course, such as which directory contains the configuration files, where lab reports should be saved, or the URL of a monitoring dashboard. These can be recorded here:

Additional notes:

-
-

A.3 | Demo Pipeline: From Camera to Browser

Big-picture idea

The demo does this:

1. **Grabs live video** from the camera.
2. **Prepares the video** in the right size/format for DeepStream.
3. **Runs PeopleNet** (a pre-trained AI model) to detect people.
4. **Draws boxes and labels** on the video.
5. **Compresses the video** (H.264) so it's small enough to send over the network.
6. **Sends the stream over UDP** to a server (beXStream®), which then can show it in a web browser.

The pipeline is implemented with GStreamer and DeepStream. In Python, it is created using `Gst.parse_launch` with the following string (line breaks added for readability):

```
nvarguscamerasrc sensor-id=0 !
  video/x-raw(memory:NVMM), width=1280, height=1080, framerate=30/1 !
  nvvidconv ! video/x-raw(memory:NVMM), format=NV12 !
  m.sink_1 nvstreammux name=m batch-size=1 width=1280 height=720 !
  nvinfer \
    config-file-path=/opt/nvidia/deepstream/deepstream-7.1/\
sources/apps/sample_apps/deepstream-test1/peoplenet/\
peoplenet_pgie_config.yml !
  queue ! nvdsosd ! queue !
  nvvideoconvert copy-hw=2 ! video/x-raw, format=NV12 !
  x264enc tune=zerolatency bitrate=2000 speed-preset=superfast \
    key-int-max=30 insert-vui=true aud=true byte-stream=true threads=2 !
  video/x-h264,profile=high,stream-format=byte-stream,alignment=au !
  rtph264pay config-interval=1 pt=96 !
  queue leaky=downstream !
  udpsink host=<JANUS_IP> port=<PORT> sync=false async=false
```

Below we explain each part of this pipeline step by step, using informal language so that students can follow even if they are not yet familiar with GStreamer.

1. Camera input

`nvarguscamerasrc sensor-id=0` selects the Jetson camera with ID 0. It outputs raw (uncompressed) video frames from the sensor.

`video/x-raw(memory:NVMM), width=1280, height=1080, framerate=30/1` is a *caps filter* (capabilities filter). It forces the camera to output:

- resolution of 1280 px $\times$ 1080 px,
- a frame rate of 30 frames/s, and
- video stored in `NVMM` (NVIDIA's GPU-friendly memory).

At this point, we have a 1280×1080 stream at 30 fps, directly in memory that is efficient for GPU processing.

2. Format conversion and stream muxing

`nvvidconv` is a video converter. It can change the resolution and the colour format of the frames.

`video/x-raw(memory:NVMM), format=NV12` forces the frames to use the `NV12` colour format, which is commonly used and well supported by NVIDIA hardware accelerators.

`nvstreammux name=m batch-size=1 width=1280 height=720` is a DeepStream element that combines one or more video streams into a batch. In this demo we only use one camera (`batch-size=1`), but the same element can later be used with multiple cameras. It also resizes the frames to 1280 px $\times$ 720 px, which is a good compromise between image quality and processing cost.

The notation `m.sink_1` simply connects the camera stream to the first input of this `nvstreammux` element.

3. AI inference with PeopleNet

`nvinfer` is the DeepStream element that runs the AI model on the GPU.

`config-file-path=…peoplenet_pgie_config.yml` points to a configuration file that tells `nvinfer` which model to load (in this case, PeopleNet), how to prepare the images, detection thresholds, and other parameters.

For each frame, `nvinfer` sends the image to the GPU, runs PeopleNet, and produces detections (bounding boxes and labels). These results are attached to the frame as metadata; the pixels of the video are not yet modified.

4. Drawing bounding boxes and labels

`queue` adds a small buffer between elements. This helps different parts of the pipeline run in parallel without blocking each other too much.

`nvdsosd` stands for "On-Screen Display". It reads the detection metadata created by `nvinfer` and draws boxes, labels, and confidence values on top of the video frames.

After this element, the video already shows visible rectangles around detected people.

5. Preparing the video for encoding

`nvvideoconvert copy-hw=2` converts the video into a format that the encoder understands. The option `copy-hw=2` tells it to use a hardware-accelerated path to improve performance.

`video/x-raw, format=NV12` ensures that the frames are raw video in `NV12` format, ready for compression.

6. H.264 video encoding

`x264enc` is a software encoder that compresses the video to H.264. Some important options are:

- `tune=zerolatency`: reduces internal buffering to make the stream more real-time.
- `bitrate=2000`: sets a target bitrate of about 2000 kbps (2 Mbps), balancing video quality and network usage.
- `speed-preset=superfast`: chooses a fast encoding preset, using less CPU at the cost of some compression efficiency.
- `key-int-max=30`: inserts a keyframe at least once every 30 frames (about once per second at 30 fps), which is useful for streaming and for recovering from packet loss.
- `threads=2`: uses two CPU threads for encoding.

`video/x-h264,profile=high,stream-format=byte-stream,alignment=au` tells the pipeline that the data is now H.264 video, using the "High" profile and a specific way of packaging the frames that is suitable for streaming.

7. RTP packaging and network streaming

`rtph264pay config-interval=1 pt=96` is an RTP payloader. It takes the H.264 frames and packs them into RTP packets, which are a standard way to send real-time media over the network. The option `config-interval=1` ensures that codec configuration is sent regularly, making it easier for receivers to join the stream. The option `pt=96` sets the dynamic payload type number.

`queue leaky=downstream` adds another buffer, but in this case old frames may be dropped if the receiver is too slow. This is important for real-time applications: it is better to show the most recent frame than to build up a long delay.

`udpsink host=<JANUS_IP> port=<PORT> sync=false async=false` sends the RTP packets using UDP to the Janus server:

- `host` is the IP address of the Janus server,
- `port` is the UDP port where Janus is listening.

The options `sync=false` and `async=false` tell GStreamer to push frames out as soon as they are available, without trying to adjust timing, which further reduces latency.

Summary

In summary, this pipeline connects the camera to the PeopleNet model through a chain of GStreamer elements. The camera provides live video, which is formatted and resized for DeepStream, processed by PeopleNet to detect people, annotated with boxes and labels, compressed as H.264, and finally sent via UDP to a server. beXStream®can then redistribute this stream using WebRTC so that you can see the AI-annotated video directly in a web browser.

